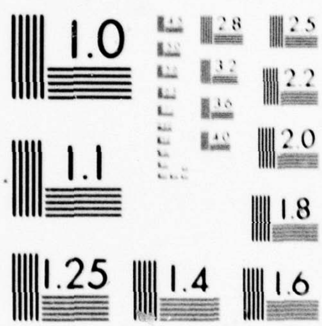
AD-A071 674 IBM THOMAS J WATSON RESEARCH CENTER YORKTOWN HEIGHTS N Y F/G 5/10
A CLINICAL-EXPERIMENTAL ANALYSIS OF DESIGN PROBLEM SOLVING.(U)
APR 78 J M CARROLL, J C THOMAS, A MALHOTRA N00014-72-C-0419
UNCLASSIFIED RC-7081 NL

1 OF 1
AD
A071674

END
DATE
FILMED
8-79
DDC


1.0
2.8
2.5
3.2
2.2
3.6
4.0
2.0
1.1
1.8
1.25
1.4
1.6

14 RC-7081 (#30341) 4/24/78
Computer Science 22 pages

LEVEL II

9 Interim

Research Report

6 A Clinical-Experimental Analysis of Design Problem Solving

10 John M. Carroll
John C. Thomas
Ashok Malhotra

11 24 apr 78

12 30p.

15 N00014-72-C-0419

AD A071674

IBM Research Division
San Jose · Yorktown · Zurich

349250

79 07 24 080

RC 7081 (#30341) 4/24/78
Computer Science 22 pages

A Clinical-Experimental Analysis of Design Problem Solving*

John M. Carroll
John C. Thomas
Ashok Malhotra

ABSTRACT: Two studies of design problem solving are reported. Experiment 1 presents an observational study of an actual client-designer work session. Analysis of the session transcript reveals a systematically structured interaction. The client and the designer decompose the overall design problem into sub-problems, each of which is smaller and somewhat more well-structured than the overall problem. Experiment 2 is a laboratory study. The "client" role is simulated by an instruction booklet; subjects play the "designer" role. Again, it is found that subjects spontaneously structure the elements of a design problem into sub-problems the nature of which is systematically related to aspects of problem structure. There is high intersubject agreement as to how the decomposition into sub-problems should proceed.

This research was supported in part by the Engineering Psychology Programs, Office of Naval Research.

LIMITED DISTRIBUTION NOTICE

This report has been submitted for publication elsewhere and has been issued as a Research Report for early dissemination of its contents. As a courtesy to the intended publisher, it should not be widely distributed until after the date of outside publication.

Copies may be requested from:
IBM Thomas J. Watson Research Center
Post Office Box 218
Yorktown Heights, New York 10598

One approach to the study of human problem solving involves mapping the inferred stages or states that the problem solvers thinking visits in the course of solving a problem (e.g., Newell & Simon, 1972). This is often called the information processing approach. Such theorizing views the problem solver as a device that explores a problem tree systematically, in a manner not unlike that of a computing device. Indeed, artificial intelligence modelling has often been employed by information processing theorists.

As is typical in the area of problem solving, most of the research pertaining to the information processing approach has been directed at the analysis of relatively simple, puzzle-problems. In this paper, we consider the application of this theoretical program to classes of problem solving we refer to as "design". Examples of design problem solving include composing a fugue (Reitman, 1965), designing a house (Simon, 1973), writing a letter (Thomas, Note 1), creating a natural language utterance (Carroll, Note 2), and designing computer software (Malhotra, Thomas, Carroll, & Miller, 1978). It is characteristic of these problem solving activities that they are relatively complex and ill-structured (Reitman, 1965; Simon, 1973). That is, there is no problem tree representation for these problems: they are too complex; and there are many, many ways to "solve" them.

This is a rather different situation from what is typically meant by problem solving in the psychological literature. On the other hand, however, these examples of design problem solving seem far more typical of ordinary commerce with the world than, say, anagrams and theorem proving. There is no notion of an antecedently defined "goal state" in design problem solving, neither is there a small set of allowable "moves". The goal of a design effort can be only partially characterized, although given a putative solution, the designer can determine whether or not it is acceptable. The set of allowable moves may or may not be characterizable; certainly in the case of composing a fugue, it is not. Given this state of affairs, we might ask whether the information processing type analysis can be adapted to design problem solving.

Can there be a state description of the processes involved in design? Surely, there can not be as complete a description as there can be for simple puzzle-problems. However, the ill-structuredness and complexity of design does not necessarily imply that the organization of behavior in design is arbitrary. Any model of design will be limited by this complexity and ill-structuredness, but we may still hope to describe a systematic structure of stages, states, steps, or units of some kind, with their control processes. In the present study, we address the question of whether, and if so how, designers structure, or decompose, their problem solving activities into sub-problems that are presumably more well-structured and less complex (Simon, 1973).

We report two studies. The first involves a dialog between a client and a designer pertaining to an actual design problem. We develop an analysis of apparent patternings in this videotaped interaction. The second study involves an experimentally contrived design problem. The client role is simulated by a booklet, and subjects play the role of designer. We present both clinical and statistical analyses of the obtained subject protocols.

<u>EXPERIMENT 1</u>

Experiment 1 is observational. We introduced only minimal measurement manipulations into an essentially naturalistic design session, involving a client and a designer. Using the design session format allows us to obtain protocol-like accounts of the sequence of a behavioral episode without forcing participants to comment on their own behavior (Newell & Simon, 1972). Moreover, we do not need to make the rather dubious assumption that participants <u>can</u> comment with any accuracy on their own internal mental states and processes (Nisbett & Wilson, 1977).

In a design session, it is perfectly natural for client and designer to verbalize. However, in order to generalize our approach to design situations involving only a single person who is both client and designer, we must assume that something analogous to the dialog between client and designer roles goes on inside of a single person taking both roles. While this may be a useful idealization, it is almost certainly untenable in the limit.

Method

Two staff members at a scientific research center volunteered to participate in our design session. The "client" was a head librarian, and the "designer" was a systems engineer. The client had a real design problem, or a set of problems, and the systems engineer was actually expert in the problem areas. Thus, the behavior we recorded was real-world design behavior.

The two participants, who had not previously discussed the librarian's design problem, met and spontaneously proceded to attack the design problem. They were given no special instructions. In the room with them were two video cameras and a microphone. The experimenters operated video tape recording equipment from an adjacent room. No one was in the room with the participants who were free to procede with their design session as they pleased.

The entire session took 35 minutes. (Full transcripts can be found in Appendix 1.)

Analysis and Discussion

At one level of analysis, we find the design session to consist of a series of what we shall call <u>cycles.</u> Each cycle starts with the client introducing some requirement, or set of requirements. The requirements addressed in the course of a given cycle are not randomly composed with each other: they typically pertain to a common sub-goal of the overall design problem. After some exploration, a sub-solution is proposed, usually by the designer. The sub-solution may then be elaborated until it is finally accepted or rejected. This completes the cycle. If a sub-solution is accepted, the next cycle starts with the client introducing further requirements. If the sub-solution is rejected, the next cycle starts with an elaboration of the requirements it was unable to meet. Usually, this elaboration includes the introduction of requirements that the client has not explicitly introduced yet or has not completely elaborated.

In order to clarify what we take to be a cycle consider the following example. This example is the third cycle (of seven) in the library design session (see Appendix 1).

> Client: ... Now, I've got some problems with where I place the printer, where the bloody control unit can go. I'd love to get the control unit buried under the floor somewhere.
>
> Designer: Uh-uh.
>
> Client: I don't know how minor that is. I think its kind of minor.
>
> Designer: The control unit can be some 2000 feet from the scope so if you have an empty closet somewhere we can sort of hide it there so long as it is accessible.

At the beginning of the cycle, the client states a requirement; he wants to reduce the clutter around the computer terminals in the library by moving their control unit. The designer offers a solution to this requirement by pointing out that the control unit can in fact be moved quite some distance away from the terminals.

The cyclic structure of design sessions can be totally sequential, one cycle after another. However, the structure can also be hierarchical (in the sense of discussing details of design elements introduced in earlier cycles): in the initial cycle of the library design session the client addresses the layout problems of the library in general. Then, in each of four successive (sub-) cycles he elaborates certain details of these problems (the example cited above is the

third of these cycles). The last of these four "embedded" cycles itself embeds two cycles. This structure of five cycles is followed by two additional cycles, making a total of seven. The structural relations between these different cycles is sketched in Figure 1 and detailed in the headings of Appendix 1.

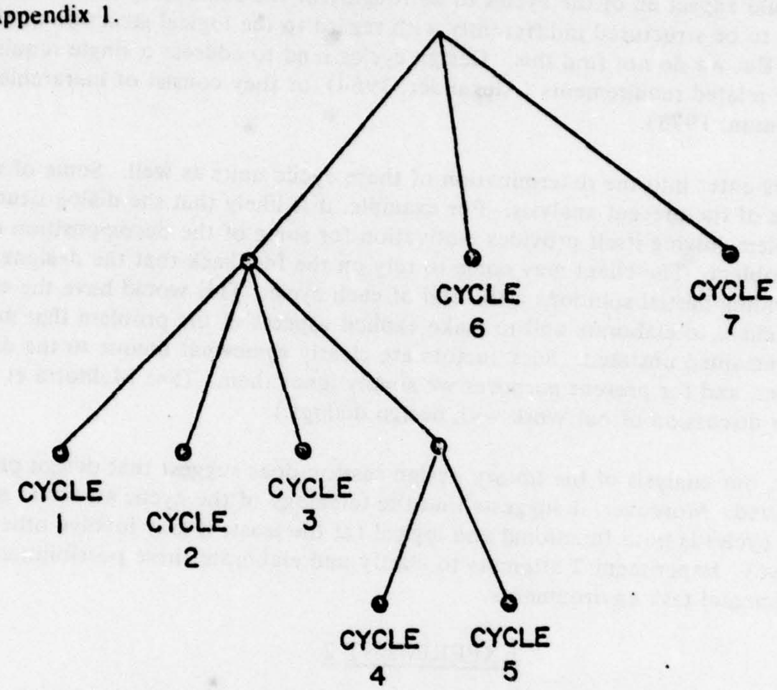


Figure 1 Caption: Structure of seven design cycles.

The relation between cycles in the structure of the design session is not, however, that of constituents in a static structure. Each cycle alters the possible substance of succeeding cycles. A given solution for cycle n, implicitly limits the range of solutions that will be considered in cycle n+1. *Indeed, the dynamic relation between cycles can operate backwards as well.* In the library session the solutions for the final two cycles (the decisions to investigate local access to system TTTT and the spooling of output to a central printer, see Appendix 1), strongly impact the solutions for the first five cycles. In effect, they raise new solutions that cover the design requirements introduced in the earlier cycles. Had the topics of the seven cycles been taken up in reverse order, the substance of each of the seven cycles would have been quite different.

Several questions are suggested by our analysis of the library design session. First, why is the session organized into cycles at all? Why doesn't the client simply list all of the requirements at the very outset of the session. It seems strange that a client, such as our librarian, who is well enough organized to hierarchically structure the first five cycles of the session, would not present the substance of all seven cycles immediately. It is stranger yet, since, as noted above, the solutions of the final two cycles alter the solutions for the first five. One hypothesis is functionally based: the limited memory and attention span of client and designer encourages them to decompose a "complex" design problem into several simpler design cycles, each concerned specifically with a part of the overall design goal.

A second question raised by our analysis is why the cycles should be structured in the particular way that they are. Some of the cycles are extremely brief (e.g., the example cited above), others are rather complicated and involve numerous exchanges between designer and client. Also, as noted above, the set of design requirements addressed in a given cycle seem always to be highly interrelated, each an aspect of a common sub-goal in the overall problem.

Prima facie, this seems to require an elaboration of the purely functional hypothesis; for if the cyclic structure of the session was entirely due to functional limitations of memory and attention, we would expect all of the cycles to be roughly of the same size, and we would expect the cycles to be structured indifferently with regard to the logical structure of the design problem. But we do not find this. Design cycles tend to address a single requirement or a set of highly related requirements (Alexander, 1964), or they consist of hierarchies of related cycles (Simon, 1973).

Other factors enter into the determination of these cyclic units as well. Some of these lie beyond the scope of the present analysis. For example, it is likely that the dialog situation of cooperative problem solving itself provides motivation for some of the decomposition of the overall design problem. The client may come to rely on the feedback that the designer provides by providing partial solutions a the end of each cycle. This would have the effect of encouraging the client to elaborate and to make explicit aspects of the problem that might otherwise have remained unstated. Such factors are clearly somewhat unique to the dialog situation, however, and for present purposes we simply ignor them. (See Malhotra et al., 1978, for further discussion of our work with design dialogs.)

In summary, our analysis of the library design session does suggest that design problem solving is structured. Moreover, it suggests that the teleology of the cyclic structure of design problem solving cycles is both functional and logical (at the least; it may involve other factors as well, see above). Experiment 2 attempts to clarify and elaborate these possibilities in more restricted experimental task environments.

EXPERIMENT 2

In Experiment 2 we "simulated" the client role by prepared written instructions, instead of merely recording an entire design dialog session. Thus, each of our subjects played the designer to a single client -- taking the form of a booklet. Essentially, we provided subjects with an unorganized list of requirements for a design problem. What we looked for is whether, and if so how, subjects spontaneously structure these requirements into design "cycles".

Method

Materials. The materials for the experiment consisted of booklets. The booklet explained to the subjects that they would be designing a schedule for a hypothetical library. The library staff consisted of ten librarians, who were not described. The work which was to be scheduled consisted of 22 tasks, each of which was briefly described. Examples of the tasks are given below. (The 22 tasks are listed in Appendix 2.)

Books that are left out in the Reading Room must be reshelved.

People who have borrowed a book for more than one month must be sent an overdue notice.

New acquisitions must be placed in the New Books display.

Subjects were also supplied with a floor-plan of the library. The instructions asked subjects to organize the library with respect to librarians and tasks.

Procedure. The ten subjects, who were staff members at a scientific research center, were run in a single group. Subjects wrote out their design solutions in any format they cared to. The experimental session took two hours.

Results and Discussion

The procedure of Experiment 2 leaves the subject quite a bit of lattitude for defining what would constitute a solution and how to go about creating such a solution. Accordingly, there

are many possible analyses of the obtained protocols and final library schedules. For present purposes, we present two types of analysis. First, we examine the "cycles" that characterize the subjects' development of a solution in time. Secondly, we examine the manner in which subjects assigned the 22 tasks to the 10 librarians in their final schedules. Finally, we comment on the relation between these two aspects of solution structure.

Cycles in the protocols. In the analysis of the design dialogues, there was a fairly clear indication of each of the phases within each design cycle (suggestion of a subproblem; consideration of various sub-solutions; acceptance/rejection; and reinitialization). This is undoubtedly partly due to the interactive turn-taking nature of dialogue. In contrast, when subjects designed a schedule for the library tasks, although they were encouraged to write down comments about what they were doing when, how they were doing it, and why, various subjects differed considerably in the degree to which they did so. For many subjects, various phases of each design cycle had to be inferred by us. However, each subject did pose and solve a number of separate sub-problems which were fairly easy to characterize.

Further, there was considerable similarity amongst subjects as to the specific content and sequencing of cycles. A complete listing of cycles is given in Appendix 3. The following general patterns were apparent. Each subject addressed one design cycle to a categorization of the tasks. Some subjects categorized the tasks several times on various different bases; for example, by area of use, by needed frequency of occurrence, by the opposition of paper work and physical movement.

A second common design cycle concerned the assignment of individuals to tasks. Nine of the ten subjects handled this explicitly. Some of the subjects assigned tasks to groups of individuals while other subjects gave each librarian a task. Some of the subjects assigned a task to an individual for all time, while others took the subproblem of determining a rotation schedule so that each librarian would eventually perform each of the different tasks.

Logically, each design design problem involves assumpttions about what is the fixed, unchangeable part context of the problem and what is the labile, changeable part. While all the subjects certainly made implicit assumptions about what could be changed in the problem, for some subjects, deciding what was problem and what was unchangeable context involved a separate cycle. Nine of the ten subjects assumed that the 22 tasks that we provided constituted an exhaustive set. One subject however began by deciding what was needed a priori in a library and after analysis added one task and subsequently categorized these 23 tasks.

The particular choice of sub-problems that subjects choose depended upon the outcome of attempting to solve earlier problems. But the manner in which this is done is not so simple as implied by a hierarchical goal structure. Consider, for example, the feasibility of running the library with ten librarians. Only one subject had an explicit design cycle addressed to tests of feasibility. Yet, two other subjects explicitly in the course of other cycles calculated the time available for work and the time necessary to do the work. Undoubtedly, had these numbers turned out so that it was impossible for the ten librarians to do the task, their problem solving would have taken a different turn. In other words, local conditions within a design cycle can produce data which in turn drive the course of the problem solving process.

The observation that people's problem solving behavior is sensitive to local results in this manner was made by Greeno (1974) in comparing the behavior of the General Problem Solver (Ernst & Newell, 1969) to the behavior of human subjects. Most of our subjects apparently had something like the following goal structure in mind: Schedule the library (categorize tasks, group people, assign groups of tasks to groups of people, map this grouping to a time schedule). However, the actual observed cycles of design varied depending upon incidental factors that arose during the design process.

One subject, for example, after assigning people to tasks seemed to 'realize' that people would get bored and then begin to provide for a mechanism to allow people to voluntarily and democratically switch tasks around. However, a consideration of this subproblem made obvious the fact that there would be conflicts of interest and therefore a mechanism would have to be evolved to deal with THAT problem. Several subjects listed assumptions in what was clearly meant to be an exhaustive list. After categorizing tasks and attempting to assign people to task, the subjects THEN realized that a whole set of other assumptions would be necessary for their assignment to be reasonable.

The general point is that the design process is an evolving, dynamic one. It is for this reason that we use the term 'cylces' rather than 'sub-problems' since the latter has unfortunately acquired the connotation that the sub-problems are logically derivable from the overall problem. Instead, we view the particular cycles as evolving dialectically from an a priori goal structure interacting with the results of successive design cycles (cf. Jeffries, Polson, & Razan, 1977). This does not mean that there is no regularity or predictability in the design process. On the contrary, despite the widely varying professional backgrounds of our subjects, there was considerable commonality both in the cycles of the design process and in the structure of final solutions.

Classifying tasks in the final solution. We now turn to characterizing the manner in which subjects organized the 22 tasks in their final solutions. Nine of the ten subjects spontaneously organized the 22 tasks into smaller sub-groups of tasks. (The remaining subject merely distributed the 10 librarians across the 22 tasks, each librarian assigned to each task for 8/22 hours per day.) The average number of sub-groups in the protocols of these nine subjects was five (i.e., 4.4 tasks per sub-group). There was considerable uniformity as to which tasks were organized into common sub-groups. When two tasks were organized into a common sub-group, they tended to be organized into a common sub-group by all, or nearly all, subjects. And when two tasks were organized into different sub-groups this too tended to be true of all, or nearly all, of the design solutions.

For each pair of subjects, we tabulated the total number of times both had organized any given pair of tasks into a common sub-group, the total number of times both had organized any given pair of tasks into different sub-groups, and the total number of non-agreements (cases in which one subject organized two tasks into a common sub-group and the other did not). This resulted in a 2 X 2 contingency table for each of 36 possible pairings of our nine subjects. Each of these tables was summarized as a chi-square value (with one degree of freedom).[1] A significant value of chi-square always indicated significantly more agreement then disagreement. Unfortunately, there is no statistical method for summarizing the overall significance of these 36 chi-squares. Since each subject is counted in 8 of the chi-squares, these chi-squares are not mutually independent. The mean obtained value for chi-square among these 36 is 10.53, and 23 of the 36 individual chi-squares exceed the .05 level. One of the nine subjects contributed 8 non-significant chi-squares. Ignoring the data from this subject, the mean value for chi-square is 13.37, with 23 of the 28 individual chi-squares exceeding conventional levels of significance.

What we conclude from the analysis up to this point is that there is some degree of agreement between subjects, first, that the overall design problem is best solved by initially decomposing it into smaller sub-problems (i.e., design "cycles"), and second, that the solution to the problem involves categorizing the tasks, which subjects do in a consistent and restricted number of ways. (i.e., certain tasks belong together, and others do not).

In order to obtain a qualitative assessment of our subjects' organization of the library tasks, we employed cluster analysis (only eight subjects were included, we deleted the subject who had been discrepant in the chi-square analysis). We defined the distance between any

two tasks to be inversely proportional to the frequency with which the two tasks were grouped together in our subject protocols. The Jardine-Sibson method of overlapping clusters, as implemented by Rohlf, Kishpaugh, and Kirk (1976), was used to recover structure in the grouping frequencies. The clustering solution we obtained for $k=2$ was relatively good. The cophenetic correlation between the k-ultrametric matrix of the clustering algorithm and the data matrix is .626, which according to the method suggested by Sokal and Sneath (1963: 315) is significantly different from zero for $p < .01$ ($z = 2.73$). We obtained two clusters with three or more members (listed by number below, see Appendix 2 for descriptions of tasks):

Cluster I: 4, 5, 7, 9, 10, 12, 18, 19, 21, 22
Cluster II: 1, 2, 3, 11, 14, 15

The tasks in Cluster I seem to pertain to the circulation desk and the card catalog. The tasks in Cluster II pertain to the reading room and the new books display.[2]

There is both a quantitative and a qualitative basis to organization of the library tasks. Subjects seem to structure their solutions into classes of tasks (nine out of ten did), and this decomposition appears both to be regular (subjects have high agreement among themselves) and to be based directly on particular salient aspects of the thematic, or logical, structure of the problem.

The cycles of the design process and the structure of the final solution are related in a complex, subtle way. The results of the design cycles help determine whether a particular attempted organization works well enough to be part of the final solution. For example, one subject's first category of related tasks was shelving new books. But this was rejected as having too few tasks (one). Conversely, the attempted organizations of the final solution will influence the design cycles. Several subjects made categorizations of tasks that turned out, on examination, not to be mutually exclusive. This fact helped provide the impetus for the next design cycle which was concerned with resolving what to do with doubly categorized tasks.

In a separate experiment, we attempt to provide more quantitative measures of the structure of successive attempted solutions to a design problem and control the cycles of the design process by controlling the presentation of information (Carroll, Thomas, & Miller, 1978). This procedure, in turn, allows a more quantitative assessment of the relationship between design cycle structure and the structure of the solutions.

CONCLUSIONS

In the introduction to this paper, we proposed a neo-structuralist program for research into the area of complex problem solving. We asked, in particular, whether designers structure, or decompose, their very complex and ill-structured problems into sub-problems. And we asked what these elements of design problem solving are like. On the basis of the present studies, we can now offer some preliminary answers, some speculations, and many, many questions.

Designers do appear to structure their problems into less complex and more well-defined elements. This structuring probably goes on at a number of levels, but we have begun to identify two in the present report. First, designing activity seems to address problems through the vehicle of a series of cycles: a sub-problem is posed, (sub-)solutions are investigated, accepted or rejected, and the process begins again -- until the overall problem has been treated comprehensively. Thus, the problem is decomposed in time. Second, though, problem elements are grouped (i.e., the 22 tasks of Experiment 2) into subsets. Thus, even in the final overall solution the problem is not treated wholistically.

The results of any given design cycle can profoundly influence the status of the results of any other design cycle. A subsequent cycle depends necessarily on the results of the cycles

that precede it, they determine what aspects of the problem will be developed, how, and in what order and priority. But subsequent cycles can also affect the results of earlier cycles -- they can obviate the pertainence of earlier cycles and cause their results to be discarded in the final solution. Analogously, the decision to assign a subset of problem elements to a common subgroup delimits what further assignments will be made concerning other sub-groups.

Further case studies will be required to assess the empirical and theoretical efficacy of the program outlined here, as a program for studying design. In any case, it seems that information processing approaches to the study of problem solving must be extended to incorporate descriptions of ill-defined classes of problem solving, like design. Whether or not a coherent neo-structuralist analysis of such problem solving activities into basic behavioral and cognitive elements is feasible remains an open and interesting question.

Reference Notes

1. Thomas, J.C. An Analysis of Letter Writing. Paper presented to the American Psychological Association, August 1978.
2. Carroll, J.M. Defining designing. In preparation.

References

Alexander, C. The Synthesis of Form. Cambridge, Mass.: Harvard University Press. 1964.

Carroll, J.M., Thomas, J.C., & Miller, L.A. Aspects of solution structure in design problem solving. IBM Research Report, RC 7078, 1978.

Ernst, G. & Newell, A. GPS: a case study of generality in problem solving. New York: Academic Press, 1969.

Greeno, J. Hobbits and orcs: acquisition of a sequential concept. Cognitive Psychology, 1974, 6, 270-297.

Jeffries, R., Polson, P., & Razan, L. A process model for missionaries-cannibals and other river-crossing problems. Cognitive Psychology, 1977, 9, 412-440.

Malhotra, A., Thomas, J.C., Carroll, J.M., & Miller, L.A. Cognitive processes in design. IBM Research Report, 1978.

Newell, A., & Simon, H.A. Human problem solving. Englewood Cliffs, New Jersey: Prentice-Hall, 1972.

Nisbett, R. & Wilson, T. Telling more than we can know: verbal reports on mental processes. Psychological Review, 1977, 84, 231-259.

Reitman, W. Cognition and thought. New York: Wiley, 1965.

Rohlf, F.J., Kishpaugh, J. & Kirk, D. NT-SYS: numerical taxonomy of multivariate statistical programs. State University of New York: Stony Brook, N.Y. 1976.

Simon, H.A. The structure of ill-structured problems. Artificial Intelligence, 1973, 4, 181-201.

Sokal, R., & Sneath, P. Principles of numerical taxonomy. W.H. Freeman: San Francisco, 1963.

Footnotes

* We are grateful to Lance Miller, who arranged the design session analyzed in Experiment 1, to Vivian Clingman, who transcribed the Experiment 1 design dialogues, to Herman Friedman, who advised us on clustering methods for part of the analysis of Experiment 2, and to Martha McRea, who helped with the analysis of Experiment 2.

[1] It may appear that the chi-square assumption of independence is not satisfied by our contingency tables, since subjects who organize, say, tasks 1 and 2 into a common sub-group, and tasks 2 and 3 into a common sub-group would be expected to organize tasks 1 and 3 into a common sub-group. We did not, however, instruct our subjects to assign tasks uniquely and exhaustively to sub-groups, and in fact several subjects assigned some of the tasks to more than one sub-group, or to no sub-group. Hence, the assumption of independence is valid for these chi-squares.

[2] There were seven clusters produced that had only two members each. These seem somewhat trivial as indications of structure in the grouping data. Accordingly, we make nothing of them in our argument. For the record, they were: 8 & 15; 7 & 20; 6 & 20; 15 & 17; 1 & 14; 15 & 16; and 1 & 6.

Appendix 1

Design Transcript: Certain identifiers have been changed in this dialog for confidentiality. Indented headings identify and describe the various cycles isolated in our analysis. J = Librarian; B = Engineer.

J: OK. The library. You know the reference area as you walk in and come across the cross aisle.

B: Right.

J: Then you go through what's the reference area into the main reading room.

B: Uh-uh

J: Kinda like watch this. The design is kind of important in what I want to do. This is the cross aisle. OK?

(DRAWS)

(CYCLES 1-5: PHYSICAL LAYOUT PROBLEMS)
(LOCAL PRINTER, CONTROL UNIT, MODEM, AND SCOPES)

You got the swinging doors. Double doors. Right? Reference desk, and people are over here. Card catalog is over here. (Right) Next to it, now, you got the scope, printer, modem underneath and the control unit is down here. We stack the people up and down. This side. Ok. Youve got four bodies sitting here (Right) -- full time.

Now its a crummy arrangement for four people. There are a number of problems involved in this. One is, what are we going to do to physically house these four people. Ive got to get some one who knows how to draw a little bit better than this to come out and give me a design that will move the card catalog (All right) and the people on this side and come out with a counter over here which will house two people and a counter on this side that will house two people. (Right) My scale is terrible, but there'll be enough room to come in. Generally, this is it. all right? I've got bodies here. Problem number 1, the simplest part of the problem, is moving all this jazz here. The printer ...

B: Right.

J: the modem (Right) and the scope. The scope is also hard-wired to the X machine downstairs.

B: OK. We dont have any problem with moving scopes around on the turret (Now) because of cable connections. (Now control unit)

(CYCLE 1: LOCATION OF CONTROL UNIT.)

J: This guy is the control unit. Now what I am thinking now is that this guy has to be moved to say here.

B: Is that to be just for your people or for user inquiries?

J: Both. Both. (hum) Right. B: So it has to be available to everybody.

Now this part of the move, I realize, is not particularly difficult. (Right) Now what I see as being a problem is that we are are bringing in a new library system. That goes DDDD...which we are buying from the Federal Republic of Germany. ... (Hm hm) Corporate is footing part of the bill with SEs...and its... getting into a real mickey mouse game. Which system thats going to come up on I dont know. They're busy looking at the systems. I dont know whether they want to bring it up on mass storage that dies every 15 minutes or whether they want to bring it up on the T machine. Or what they are going to do. Reguires a slightly different monitor which we have here in the building but still an unproved monitor. (Right)

(CYCLE 2: NEED FOR MORE SCOPES.)

Now we get complicated. Now I need another scope, ok, to handle what that system's gonna do. See this does not talk to ... Yes, this does talk to downstairs. It also has to be capable of talking to that DDDD file, lets call that D. It also talks to I, which is TTTT. (Right) Now.

B: Now, how does TTTT ... ok, so you talk to TTTT right now. (hum) Do you also talk to our system downstairs?

J: No.

B: Strictly, (Strictly) strictly TTTT?

J: It strictly goes to TTTT.

(CYCLE 3: LOCATION OF PRINTER.)

Now, I've got some problems with where I place the printer, where this bloody control unit can go. I'd love to get the control unit buried under the floor somewhere

B: Uh-uh. All right. But that's that.

J: I dont know how minor that is. I think its kind of minor.

B: Right because the control unit can be (hum) some 2000 feet from the scope so if you have an empty closet somewhere we can sort of hide it there (OK) so long as it is accessible to the SE's.

(CYCLES 4 & 5: CONTROL UNIT AND MODEM CAPABILITIES.)

J: Alright now. The scope that talks to the DDDD which could be this one because we are bringing it up here (Hum all right?) After two years it will go to headquarters. Where in headquarters it will go to I don't know. So now what I'm gonna want to do is add a scope, and I want to add a scope now. One more scope. (Right) I want the scope here so I can drop a line downstairs to talk to DDDD. (m hum) But the same time I want to know, ok, now in two

years when this goes, this equipment I have now, the control unit and the modem, all right? Can it handle two scopes. Can it handle two scope that might go in two different directions. In other words, theres Arlington, headquarters may take this DDDD thing, plant it down in Belmont, not talk to the same machine. Can this control unit take care of both of these guys, the one that goes to D and the one that goes to I.

B: Ok, uh I ..... (muttering)

Now let me understand the TTTT connection right now taik about it. Explain the TTTT connection to me now. Do you have a modem that is hardwired into a system in a system in White plains right now?

J: A leased line comes through.

B: A leased line.

J: A leased line from the control unit to the modem.

(CYCLE 4: MULTIPLE LINES INTO MODEM.)

B: We may have to consult the sales manual. (Pulls out manual) (Let's do that) I'm stuck, right off the top of my head, to give you an answer. (flips pages) You're using a xxnn?

J: Yes. Scope is... xxnn...

B: nm? J: nm. Model 1 control unit.

INTERRUPTION

Lance: We had to stop for about 5 minutes while B consulted the reference books on the equipment to get the answer to J's question. We now pick up as B is just finishing searching for the answer.

SESSION RESUMES

B: Ok. We have a feature that can be put on the 3872 modem that's called "fanout" that will allow us to attach multiple lines to the modem. And it says here that as long as the machines are used one at a time you and should have no problem with the lines as long as you don't have 3 or 4 terminals trying to use the same line (ok) at the same time.

J: Alright, the modem ...
I've been asking the wrong thing. The modem doesn't particularly upset me ---having to have multiple modems. (Uh-hum) They don't take up any physical space. There's no real difficulty with those little guys. (Right)

(CYCLE 5: CONTROL UNIT AND DIFFERENT SYSTEMS.)

The control unit's what my major concern is. Can the control unit go two different directions or do I have to keep adding control units every time I add a scope?

B: Well, the control unit that you presently have is capable of being modified to handle up to 32 scopes.

J: Ah-ha. Hopefully (and they) ... potentially, in different directions?

B: Different directions would mean -- ah -- possibly more modems or the additional feature on your present modems to go in different directions.

J: Ok. But the control unit, essentially, can stay as a single unit? I don't need multiples on that?

B: That's right. (ok) You can increment that in increments of four features, four scopes at a time.

(CYCLE 6: SPOOLING OUTPUT TO CENTRAL PRINTER.)

J: Ok. That means I am not running into a problem on that yet. Ok. Second part of my question is : How feasible is it right now, (Um) with what I've got that talks to I, which is TTTT, which is the setup we have now, that I mentioned went through, first ...

B: Right.

J: ... to fool the hardware that we have, since getting programming support out of the people down at TTTT is difficult. Right now (Yeah) a guy that's sitting here and doing searching on AAAA using the scope.

B: Alright.

J: If he sees any citations that he likes then he says "Ok. I'd like to have that printed. I don't want to sit here and write." He has to go through this copy routine on the scope. He's in copy it says "Ok. Identify the printer". Identifies the printer and then it takes forever as this (bang bang) printer clanks away. (We) Can we fool it and let it go downstairs and spool it off rattle it offf on the high-speed printer. So that while now the guy is pretty-much limited. So if he has 5 or 10 things he is willing to wait ten minutes for it to print. Fine. If the guy has 100 citations he's got to tell them to do a batch print and wait a week till the thing finally arrives. Can we fool it? Spool it off save it zap it downstairs, here?

B: That could be possible. Ok. I'd have to consult software people in order to answer that question. But ...

J: It can't be solved by hardware? But I ...

B: Ah. We have another alternative. We can change the printer. I think you have the kkkk printer right now which works right off the scope's screen itself.

J: Well. It's not it's not screen dependent.

B: Ok. But you have to issue the command to make (that's) the scope display the printout. You have to issue the copy command ...

J: No. No. It's not the scope display it prints. I'm looking through a file, Right Ok, ...

B: Right.

J: ... and I see a document, for instance, and it may take up the whole screen itself.

B: Right.

J: It'll tell you on the top that it is document 1 of, lets say 100 ...

B: Right.

J: ... or better yet, say its 99 of 100. I can go copy 1, which means copy document 1, that you have showed me earlier in this game queue.

B: Uh-uh. And where is that copy printed?

J: Its read on the local printer and I keep on doing whatever I want to do on the scope independent of the printer.

B: Ok. There is.

J: I turn the printer off, physically turn the printer off, cause I cant stand the clanking, turn it back on again and it keeps right on going. It picks up right where it left off.

B: O.K. that.

J: Takes it, spools it out, sends it off to the ...

B: Right. I know that's a problem, the clanking, with this type of printer, it's a noisy device.

J: It doesnt clank, it buzzes.

B: Its a noisy device. In a limited area, like the one we're talking about right now, its hard to work with. (Yep) I understand that. There's another type of printer, there's a llll printer which is ... It would give you every line of input and output simultaneously -- Instead of you asking for a particular line to be printed, you would get everything and then if you asked ...

J: I don't follow you.

B: O.K. Every entry that you make on the scope (humh) will be printed on the llll. Every answer that you receive on the scope, or every screenful of information that you get (humh) for an inquiry, will print on that llll. So it is in effect a hard copy device.

J: Sentence printer.

B: Right. Now, that might be easier to go to but like I said before if we wanted to take your original suggestion, (that's) I would have to consult with software.

J: That's,that's gonna that other way's gonna spit paper out like there's no tomorrow. Cause then theoretically when the guy comes on he'd have to have the printer on in case he wanted something ...

B:

J: ... and then he'd have to wade through the paper and say "Ah, this is the one I want" and tear that one out.

B: Right. The printer would be on at all times.

J: O.K., No, Now that's not what I'm after. I'm after the selective way where the where the guy says "O.K. I want this ..."

B: Hmm-hum

J: And we can take the guy, he can have 100 documents, 200 documents, but have them today as opposed to waiting a week for them.

B: Right. O.K. As I said, I'd have to consult with our software people and that may be a project of you know a month to have the software written, I'm not sure.

J: O.K.

B: Now, uh,(yes) with your diagram here -- (right) I notice that it looks a little tight. -- you know we may ...

J: Well my scale is magnificent. (laughs)

B: We may want to change things around a little bit, but ...

J: No I am just theoretically sticking things in. (hmm) This is where it is, this is where the control unit is. I'm saying the scopes here and here. (humhu) Physical arrangement of the scopes up here is yeah that's open to debate and whoever has to move them will love it, solid walls and solid ceiling in that area so running wires is a problem in the beginning. (Right)

B: Right.

J: Alright, software on this printer, we'll see what can be done on that. Now, let's push that one step further.

B: Okay?

(CYCLE 7: LOCAL ACCESS TO SYSTEM TTTT.)

J: Let's push it all the way out and say, well that printer out there may be a convenience, but if we've got the control unit that talks to the folk in Arlington, why can't we get rid of the control unit and get rid of the modem and maybe let big mama downstairs fool it, so that you, for instance, if you want to access TTTT ...

B: Right.

J: ... there is a sequence you must got through, you gotta go through the protocols, you gotta give them the proper passwords.

B: Right.

J: If you B.J. want to use it, you pull your tube and you ask to use it. (hum) Once you pull the tube and you log on, you see what the machine is, you log on as B.J.

B: Right.

J: And in order to get at that lets just theoretically say o.k. you have to share my user lib which says, yes o.k. B.J. is in there, he's o.k. to use TTTT, I've given him permission.

B: Uh-uh.

J: You can use TTTT from your desk and do not have to walk up here and use the single routine in the library. Now, anything you want to come back, you say copy or print, or whatever you say to do comes back through your scope, which comes back through the userlib which says you're B.J., and comes into your folder downstairs.

B: O.K. That's a good idea. We may, we have rrrr control units downstairs. Its quite possible that we could have a line removed from the tttt control unit, upstairs (hum) installed on the rrrr on the VVVV system downstairs. If we can do that it might be possible to access TTTT through the VVVV machine -- which would make it virtually accessible from anyone in the whole research center.

J: That may be simply to protect the password. Having to go through the my userlib.

B: Right and being on VVVV, if its running on a virtual machine under VVVV, all of your ... all or any of your output can be spooled and printed at the local printer.

J: Ok. That ... that's interests me more than doing anything with the printer. If that ... if that's feasible I would like to leave the printer where it is.

B: Uh-uh.

J: Fine. Leave it alone. That's fine, if the guy's in the library and he wants a little thing and he wants it here and now ...

B: Right. Now. These scopes go through the building network to the scope control units downstairs into VVVV.

J: That says the modems can be done away with too?

B: Yeah. Then everything goes.

J: You wouldn't need either of those?

B: Right. You wouldn't need either your control unit or your modem.

J: Ok. Can we do something on it? See what we can do? Forget about the printer. Forget about the rest of this jazz and just move in the direction of having ...

B: Right. Sure.

J: ... direct output.

B: It's not going to be an overnight affair.

J: Oh, no, its not.

B: We are going to have to going to have to go through feature changes on the rrrr which could possibly take as long as six months to get the feature.

J: Ok. TTTT told me today we should upgrade the modems we have now and I have a delivery date of MM, so that that's overnight. So six months doesn't worry me.

B: (Laughs) At least we can investigate and see if its more than we can do right now.

J: You gotta issue a shop?

B: Yeah, I'll have to talk to E in Engineering.

J: Alright. Can we see what we can do with it?

B: Sure.

J: I'd like that; I'd dearly love to do it.

B: Yeah.

J: Besides anything else it would free some space that I need.

B: Right.

J: Potentially, I don't need all these scopes here.

B: Uh-uh.

J: I can do with one. Because I have scopes in the office, I have scopes in the back room. So now anybody's scope in the building can essentially do what these can.

B: That's Right. These scopes that are tied into your own system here. You can use them right on the VVVV system.

J: Now, political question is TTTT gonna get unhappy if we start spooling their ... coming out ... then are we going to steal their files.

B: (Tries to say something.)

J: But that's internal and I suppose we can set that problem up. Deal with the problem as we set this up so that its a "No save". (I) Read, write and that's it.

B: Right. Since I'm not familiar with the internals of TTTT, but I've heard that ...

J: All their data base is not going to give us a problem because its IBM documents.

B: Uh-uh.

J: The other two they subscribe to so there may be a difficulty if we read and save ...

B: Uh-uh.

J: ... unless we have it built in somewhere that we read, write and don't save.

B: Right. Right.

J: But that I can check on if we believe there's a problem.

B: Yeah. We can look into the possibility if maybe there are more people in this building that are have TTTT terminals that you are not aware of.

J: No. They don't.

B: They don't?

J: They don't. That's a guarantee. They have profiles. They get those little cards each week from ...

B: Uh-uh. (you know) But they can only access it through your terminal system.

J: And the problem with it is an obvious one. If the guy's up here and the terminal is free maybe he'll use it.

B: Right.

J: And if the guy happens to live close by and wants to use it, its not a problem but if the guy happens to be way off in the basement he's not coming up here. He's just ...

B: Or if he comes and finds the terminals busy a couple times.

J: He just doesn't come back any more.

B: Right.

J: If its convenient and its there and if its in front of him and if he can get what he wants by doing it that way he's more likely to do it.

B: Uh-uh. But we can do ...

J: The command structure on it is ridiculously simple. There are six basic commands. So that, you know, with the sophistication in this building in using terminals its not going to be a problem, an education problem.

B: Right.

J: You just hand the guy a manual and say "Bye, enjoy yourself. (yeah) If you have a problem let us know."

B: It should make life easier for your people, getting rid of the printer.

J: Yeah, its just noisy and its in the way. O.K., so if we can see what Ira will do on that one, we take care of all the other problems ... that wipes them all out.

B: Right, talk to Ellis, ...

J: Right.

B: TTTT, and an RC, right?

J: Right.

B: If we can go that far then I think we've got your problem solved. Now, if for some reason the rrrr doesn't pan out ...

J: Then were back to where we were before.

B: We're back to ...

J: The control units and the modems

B: Right. (Go on that) And this would still be probably a three to six month time frame in getting the features and ...

J: No problem. No problem. It's as I say, this is generally what I'd like to do with the reference area, but I wanted to get somebody who can, you know,

B: uh-uh

J: An interior decorator, a designer an architect, somebody who can do that properly.

B: Yeah, right, it's ah this ... These scopes won't have any noise problem, or heat problem. (hum) That's been a problem in the past.

J: Yeah, this one is ideal. It gets all the stuff out of the way.

B: O.K.

J: O.K., I think that is all we can do for today.

B: Yeah that's without the other people.

Appendix 2

Twenty-two library procedures used in Experiment 2.

(1) Books that have been in the New Books display for more than one week must be placed on the shelves;

(2) When the Magazine Rack becomes disordered, it must be reorganized;

(3) Books that are left in the Reading Room must be reshelved;

(4) The names of books that have been borrowed must be entered into the Books Borrowed listing at the circulation desk;

(5) New books that have been requested by someone must be purchased by a mail order sent to the publisher;

(6) Books that have been returned must be replaced on the shelves;

(7) Books that are currently stored in the archives that have been requested by someone must be moved from the archives to the circulation desk;

(8) Unbound periodicals that have been on the magazine racks for more than one week must be moved to the magazine stacks at the circulation desk;

(9) People that have borrowed a book for more than one month must be sent an overdue notice;

(10) When a book is moved to the archives, the entry for it must be changed in the Card Catalog to tell library users where the book currently is;

(11) Magazines that are left in the Reading Room must be replaced on the magazine rack;

(12) When a book is retrieved from the archives, the entry for that book in the Card Catalog must be changed;

(13) Notices must be sent out to library users who have requested new books when those books arrive;

(14) New acquisitions must be placed in the New Books display;

(15) New unbound periodicals that arrive in the mail must be placed on the magazine rack;

(16) New bound periodicals must be placed on the shelves;

(17) Unbound periodicals that are more than one year old must be collected and sent to the bindery to be bound;

(18) New acquisitions must be entered into the Card Catalog;

(19) Books that have been returned must be checked off of the Books Borrowed listing at the circulation desk;

(20) Books that have not been borrowed for more than two years must be moved to the archives;

(21) Periodicals that come back from the bindery must be entered into the Card Catalog;

(22) Notices must be sent out to library users who have requested books from the archives when the books they have requested are retrieved.

Appendix 3

Cycle structure from subject protocols, Experiment 2.

Below is a hierarchical list of the cycles of the various subjects. Omitted are cycles relating primarily to communicating about the design solution. Clearly, in what follows some judgement is required concerning the appropriate level of sub-problems to qualify as a cycle. The notion of cycle includes the notion that a goal exists which is not currently being met and no pre-existing algorithmic method exists for achieving the goal. Thus, categorizing the tasks is a cycle while alphabetizing the tasks would not be considered a cycle. Despite these definitional restrictions, considerable latitude persists in interpretting the structure of sub-problems. For example, suppose a subject first categorizes tasks, then makes groups of people and finally assigns the groups of people to the groups of tasks. Perhaps these activities mirror exactly the three sub-problems that the subject was working on in the order that they were attacked. Alternatively, the subject may have first worked on the problem of attempting to assign people to tasks mentally and in so doing realized that categorizing people and tasks first would be required. Thus, the exact structure of the sub-problems worked on is somewhat subjective. Nevertheless, the apparent similarity between subjects is instructive.

Subject One:

Design a Library Schedule
- Classify Tasks by Priority and Needed Frequency
  - By priority

By Needed Frequency
Classify Tasks by task requirements
Assign People to Tasks
Provide for Changes to the People-Task Assignment
Provide for a General Structure of Rules by which Library can Run

Subject Two:

Design Library Schedule
Categorize Tasks
Split Context from Problem
Assign People to Tasks
Split Context from Problem

(Comment: the subject listed several assumptions in an attempt to split context from problem. S then attempted to assign people to task and apparently realized that more assumptions would have to be made before this could be done.)

Subject Three:

Design a Library Schedule
Determine the Frequency of Tasks
Assign People to Tasks
Split Context from Problem
Assign People to Tasks
Split Context from Problem
Place People and Tasks within Time Schedule

Subject Four:

Design Schedule for Library
Split Context from Problem
Identify Critical Tasks
Assign People to Tasks
Assign most Critical Tasks
Assign least Important Tasks
Assign Moderate Important Tasks

(Comment: that logically one could also assume a somewhat different subproblem structure; viz., that identifying critical tasks was really subordinate to assigning people to the most critical tasks. However, it appears from the subject's comments that, at least consciously, the critical tasks were identified first for several purposes included priority of training.)

Subject Five:

Design Library Schedule
Categorize Tasks
Find tasks relating to task 1
Find tasks relating to task 2
Find tasks relating to task 4

(Comment: The subject's essential strategy for categorizing tasks was to go through the tasks in numerical order and to find the previously uncategorized tasks related to the given task.)

Find tasks relating to task 5
Find tasks relating to task 7

Find tasks relating to task 10
Resolve tasks that should be in two categories

(Comment: This subject rejected the potential subproblem of assigning people to the categories produced. This subject did not feel there was enough information to determine how many hours each task would take).

Subject Six:

Design a library schedule
- Determine the frequency of various tasks
- Determine the amount of time available for doing tasks
- Schedule the tasks on a time basis
- Assign people to tasks
  - Categorize the tasks

(Comment: It seems that time ran out before people could be assigned to the task categories).

Subject Seven:

Schedule for library
- Determine frequency of tasks
- Categorize tasks (by area)
- Determine number of hours available in week
- Determine number of shifts
- Assign tasks to people
- Determine mapping of task-people combinations to time schedule

Subject Eight:

Schedule for library
- Determine kinds of tasks
- Determine areas of library
- Categorize tasks
- Assign groups of people to categories of tasks
- Determine frequency with which tasks should be done

Subject Nine:

Schedule for library
- Determine difficulty of the tasks
- Compare this task to 'real-world' analogue and note differences
- Determine feasibility
  - Determine man-hours available
  - Determine man-hours needed
- Assign people to tasks to equalize time required
- Determine rotation schedule for remaining tasks
- Determine rotation schedule for ALL tasks

(Comment: this latter goal was expressed by the subject but time ran out before it could be implemented).

Subject Ten:

Design a Schedule for the library
- Determine list of tasks to be used
  - Make a priori list of tasks

Compare with given list
Add (one) additional task as needed
Determine logical flow of work
Determine where numbered tasks are to be done
Determine sequence of tasks within an area
Categorize and sequence tasks
Assign people to sequences of tasks

OFFICE OF NAVAL RESEARCH, CODE 455
TECHNICAL REPORTS DISTRIBUTION LIST

Director, Engineering Psychology
Programs, Code 455
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217 (5 cys)

Defense Documentation Center
Cameron Station
Alexandria, VA 22314 (12 cys)

Dr. Robert Young
Director, Cybernetics Technology Office
Advanced Research Projects Agency
1400 Wilson Blvd.
Arlington, VA 22209

Col. Henry L. Taylor, USAF
OAD (E&LS) ODDR&E
Pentagon, Room 3D129
Washington, D.C. 20301

Office of Naval Research
International Programs
Code 102IP
800 North Quincy Street
Arlington, VA 22217 (6 cys)

Director, Information Systems
Program, Code 437
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217

Commanding Officer
ONR Branch Office
ATTN: Dr. J. Lester
495 Summer Street
Boston, MA 02210

Commanding Officer
ONR Branch Office
ATTN: Dr. Charles Davis
536 South Clark Street
Chicago, IL 60605

Commanding Officer
ONR Branch
ATTN: Dr. E. Gloye
1030 East Green Street
Pasadena, CA 91106

Dr. B. McDonald
Office of Naval Research
Scientific Liaison Group
American Embassy, Room A-407
APO San Francisco 96503

Director, Naval Research
Laboratory
Technical Information Division
Code 2627
Washington, D.C. 20375 (6 cys)

Office of the Chief of Naval
Operations, OP987P10
Personnel Logistics Plans
Department of the Navy
Washington, D.C. 20350

CDR Paul Nelson
Naval Medical R&D Command
Code 44
Naval Medical Center
Bethesda, MD 20014

Director
Behavioral Sciences Department
Naval Medical Research Institute
Bethesda, MD 20014

Dr. George Moeller
Human Factors Engineering Branch
Submarine Medical Research
Laboratory
Naval Submarine Base
Groton, CT 06340

Bureau of Naval Personnel
Special Assistant for Research
Liaison
PERS-OR
Washington, D.C. 20370

Navy Personnel Research and
Development Center
Management Support Department
Code 210
San Diego, CA 92152

Naval Training Equipment Center
ATTN: Technical Library
Orlando, FL 32813

Human Factors Department
Code N215
Naval Training Equipment Center
Orlando, FL 32813

Dr. Alfred F. Smode
Training Analysis and
Evaluation Group
Naval Training Equipment Center
Code N-OOT
Orlando, FL 32813

Dr. Gary Poock
Operations Research Department
Naval Postgraduate School
Monterey, CA 93940

Dr. A. L. Slafkosky
Scientific Advisor
Commandant of the Marine Corps
Code RD-1
Washington, D.C. 20380

Mr. J. Barber
Headquarters, Department of the
Army, DAPE-PBR
Washington, D.C. 20546

Dr. Joseph Zeidner
Director, Organization and
Systems Research Laboratory
U.S. Army Research Institute
5001 Eisenhower Avenue
Alexandria, VA 22333

Dr. Edgar M. Johnson
Organization and Systems
Research Laboratory
U.S. Army Research Lab
5001 Eisenhower Avenue
Alexandria, VA 22333

Technical Director
U.S. Army Human Engineering Labs
Aberdeen Proving Ground
Aberdeen, MD 21005

U.S. Air Force Office of Scientific
Research
Life Sciences Directorate, NL
Bolling Air Force Base
Washington, D.C. 20332

Dr. Donald A. Topmiller
Chief, Systems Engineering Branch
Human Engineering Division
USAF AMRL/HES
Wright-Patterson AFB, OH 45433

Lt. Col. Joseph A. Birt
Human Engineering Division
Aerospace Medical Research
Laboratory
Wright Patterson AFB, OH 45433

Air University Library
Maxwell Air Force Base, AL 36112

Dr. Alphonse Chapanis
The Johns Hopkins University
Department of Psychology
Charles & 34th Streets
Baltimore, MD 21218

Dr. Arthur I. Siegel
Applied Psychological
Services, Inc.
404 East Lancaster Street
Wayne, PA 19087

Dr. Gerson Weltman
Perceptronics, Inc
6271 Variel Avenue
Woodland Hills, CA 91364

Dr. J. A. Swets
Bolt, Beranek and Newman, Inc.
50 Moulton Street
Cambridge, MA 02138

Dr. H. Rudy Ramsey
Science Applications, Inc.
40 Denver Technological
Center West
7935 East Prentice Avenue
Englewood, Colorado 80110

Dr. Meredith Crawford
5606 Montgomery Street
Chevy Chase, MD 20015

Professor Douglas E. Hunter
Defense Intelligence School
Washington, D.C. 20374

Dr. James H. Howard
Catholic University
Department of Psychology
Washington, D.C. 20064

Dr. Jesse Orlansky
Institute for Defense Analyses
400 Army-Navy Drive
Arlington, VA 22202

Journal Supplement Abstract
Service
American Psychological
Association
1200 17th Street, N.W.
Washington, D.C. 20036 (3 cys)

Dr. William A. McClelland
Human Resources Research Office
300 N. Washington Street
Alexandria, VA 22314

Dr. Tom Love
General Electric Company
Information Systems Programs
1755 Jefferson Davis Highway
Arlington, VA 22202

Dr. David J. Getty
Bolt, Beranek & Newman
50 Moulton Street
Cambridge, MA 02139

Director, Human Factors Wing
Defence & Civil Institute of
Environmental Medicine
Post Office Box 2000
Downsville, Toronto, Ontario
CANADA

Dr. A. D. Baddeley
Director, Applied Psychology Unit
Medical Research Council
15 Chaucer Road
Cambridge, CB2 2EF
ENGLAND

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) A Clinical-Experimental Analysis of Design Problem Solving		5. TYPE OF REPORT & PERIOD COVERED Interim Technical Report
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) John M. Carroll, John C. Thomas, Ashok Malhotra		8. CONTRACT OR GRANT NUMBER(s) N00014-72-C-0419
9. PERFORMING ORGANIZATION NAME AND ADDRESS International Business Machines, T.J.Watson Research Center, P.O.Box 218, Yorktown Heights, N. Y. 10598		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS NR-197-020
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research, Code 455, Arlington, Virginia		12. REPORT DATE
		13. NUMBER OF PAGES
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Two studies of design problem solving are reported. Experiment 1 presents an observational study of an actual client-designer work session. Analysis of the session transcript reveals a systematically structured interaction. The client and the designer decompose the

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. (cont.)

overall design problem into sub-problems, each of which is smaller and somewhat more well-structured than the overall problem. Experiment 2 is a laboratory study. The "client" role is simulated by an instruction booklet; subjects play the "designer" role. Again, it is found that subjects spontaneously structure the elements of a design problem into sub-problems the nature of which is systematically related to aspects of problem structure. There is high intersubject agreement as to how the decomposition into sub-problems should proceed.